SAND 2011-1137C

Evaluation of Mixed Continuous-Discrete Surrogate Approaches

Patricia Hough, Quantitative Modeling and Analysis
Laura Swiler, Optimization and Uncertainty Quantification
Sandia National Laboratories

Herbert Lee
Applied Mathematics and Statistics
University of California, Santa Cruz

Curtis Storlie
Statistical Sciences Group
Los Alamos National Laboratory

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Motivating Example: Weapons safety

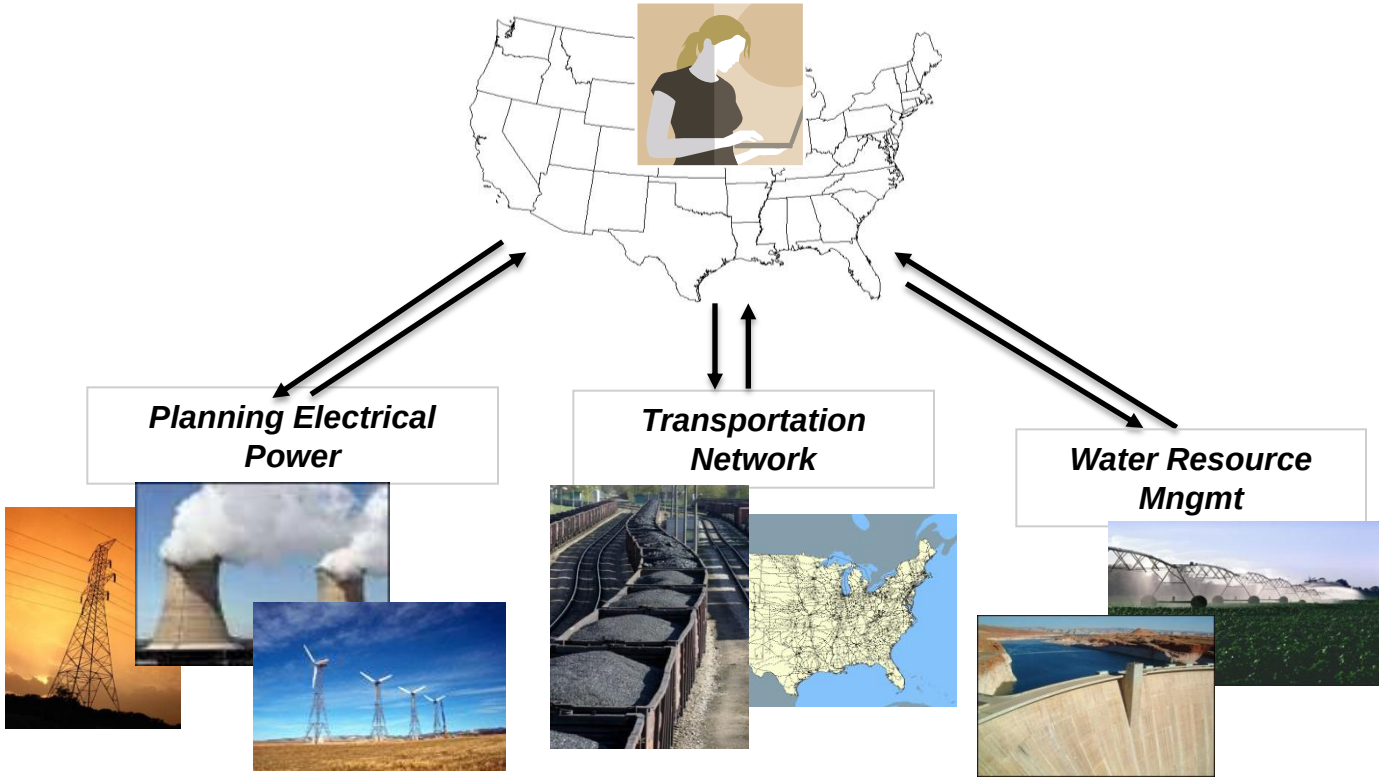
•http://safetycampus.files.wordpress.com/2008/12/

- Need to assess the Probability of Bolt Failure under tensile load
- ADAGIO implicit nonlinear dynamics code used for simulation
- **Discrete materials models and thermal behavior**
- Performance threshold:

  P(bolt failure) = 10^{-6}
- Performance margin:

  10^{-6} – P(max observed strain – critical strain > 0)
- Uncertainties: material model parameters, critical strain

Motivating Example: System of Systems



- Large–scale models are decomposed into constitutive system models
- Black-Box model of actor behavior, independent subsystems making choices based on inputs from others
- Involves mix of discrete choices and continuous design parameters

Why do we need surrogates?

- **Optimization and UQ methods (both aleatory and epistemic) are computationally expensive**
- **People want to perform a limited number of "true" simulations, construct surrogate, and perform analysis on the surrogate**
  - Engineering analysis: discrete choices for design options, physics "knobs" in codes
  - Logistics, SoS applications
- **Most surrogate models assume continuous inputs, and rely on some assumptions about how the output varies as the input varies**
  - With discrete variables, this no longer holds (especially if the variables are categorical vs. ordinal)
    - E.g., changing from 3 depots to 4 depots may result in fundamentally different behavior of a logistics system
    - Model A vs. Model B vs. Model C may result in very different behaviors

Possible Options for Modeling Discrete Variables (1)

- **Categorical Regression**
  - Uses indicator functions for various levels of categorical variables

    $Y=\beta_0 + \beta_1 X_1 + \beta_2 X_2$ where X_1 is continuous, X_2 is binary

    → $Y=\beta_0 + \beta_1 X_1$ for $X_2=0$, $Y=\beta_0 + \beta_1 X_1 + \beta_2$ for $X_2=1$

    → Results in 2 different models, both with slope β_1 but with intercepts β_0 and $(\beta_0 + \beta_2)$
  - Computationally expensive because each combination of discrete variables has a separate regression function
    - Large numbers of discrete variables and for large numbers of "levels" per variable => combinatorial explosion
    - Need enough samples over the continuous variables for EACH combination of discrete levels to obtain an accurate regression function

Possible Options for Modeling Discrete Variables (2)

- **Gaussian Process Models**
  - Responses at points close together in input space are highly correlated, responses at points far away are not
  - Can provide an estimate of the prediction uncertainty which can be used in optimization
  - Explicit Representation of Categorical Variables in GP
    - Qian and Wu, 2008
    - Recommend isotropic correlation structure
  - Treed Gaussian Process
    - TGP papers: Gramacy and Lee, 2008; Gramacy and Taddy, 2009
    - Allow partitions on categorical variables
    - Surrogate model made at "leaf" nodes is formed only on continuous variables
    - Options: Bayesian GP with Linear Limiting Model

Possible Options for Modeling Discrete Variables (3)

- **Adaptive COmponent Selection and Smoothing Operator (ACOSSO)**
  - Univariate smoothing spline estimate

$$\frac{1}{n}\sum_{i=1}^{n}[y_i - f(x_i)]^2 + \lambda\int_0^1 [f''(x)]^2 dx$$

Term which penalizes roughness

  - ACOSSO Estimate: f is an additive function

$$f(x) = \sum_{j=1}^{q} f_j(x_j)$$

$$f_j(1) = c_{1'} f_j(2) = c_{2'} \ldots f_j(m_j) = c_{m_j'} \sum_{x=1}^{m_j} c_j(x) = 0$$

$$\frac{1}{n}\sum_{i=1}^{n}[y_i - f(\mathbf{x}_i)]^2 + \sum_{j=1}^{q}\lambda_j\int_0^1 [f_j''(x_j)]^2 dx$$

$$\frac{1}{n}\sum_{i=1}^{n}[y_i - f(\mathbf{x}_i)]^2 + \lambda\left(w_j \sum_{j=1}^{q}\left\{\left[\int_0^1 [f_j'(x_j)]dx_j\right]^2 + \int_0^1 [f_j''(x_j)]^2 dx_j\right\}^{1/2} + \sum_{j=q+1}^{p} w_j \left\{\sum_{x_j=1}^{m_j} f_j^2(x_j)\right\}^{1/2}\right)$$

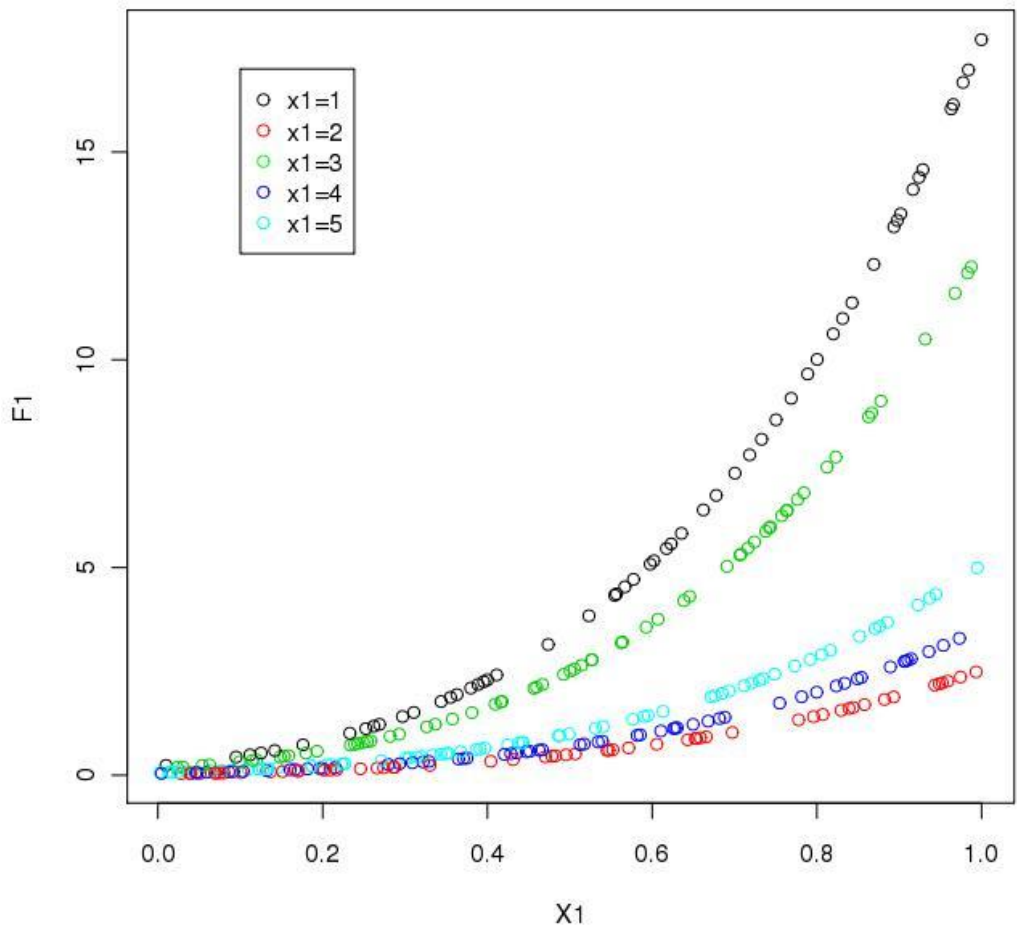
Term which penalizes trend

Term which penalizes categorical predictors

Testbed

- **Need a testbed for evaluation of these various methods**
- **Desired characteristics of a testbed:**
  - Fast running evaluations
  - Easy to compile, cross-platform compability
  - Extendable
  - File input/output
  - Scalability of function in terms of number discrete variables and/or levels per variable
  - Ability to control problem complexity

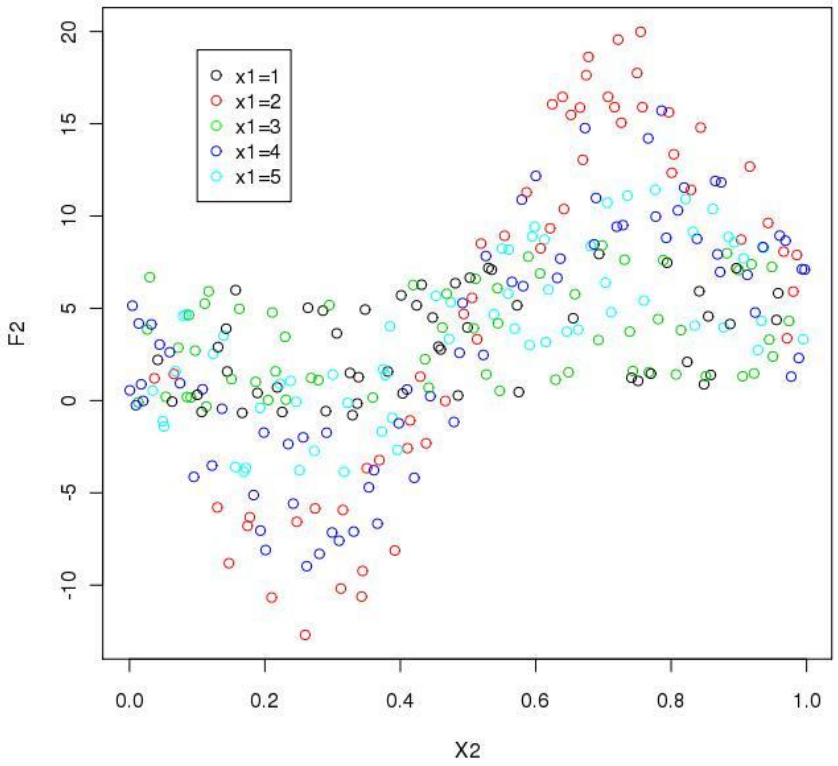
Test Function 1



$$y = f_1(x) = \begin{Bmatrix} 3.5(x_2 + 0.5)^4 \text{ if } x_1 = 1 \\ 0.5(x_2 + 0.5)^4 \text{ if } x_1 = 2 \\ 2.5(x_2 + 0.5)^4 \text{ if } x_1 = 3 \\ 0.7(x_2 + 0.5)^4 \text{ if } x_1 = 4 \\ (x_2 + 0.5)^4 \text{ if } x_1 = 5 \end{Bmatrix}$$

Test Function 2

$$y = f_2(x) = \begin{cases} \sin(2\pi x_3 - \pi) + 7\sin^2(2\pi x_2 - \pi) \text{ if } x_1 = 1 \\ \sin(2\pi x_3 - \pi) + 7\sin^2(2\pi x_2 - \pi) + 12\sin(2\pi x_3 - \pi) \text{ if } x_1 = 2 \\ \sin(2\pi x_3 - \pi) + 7\sin^2(2\pi x_2 - \pi) + 0.5\sin(2\pi x_3 - \pi) \text{if } x_1 = 3 \\ \sin(2\pi x_3 - \pi) + 7\sin^2(2\pi x_2 - \pi) + 8\sin(2\pi x_3 - \pi) \text{ if } x_1 = 4 \\ \sin(2\pi x_3 - \pi) + 7\sin^2(2\pi x_2 - \pi) + 3.5\sin(2\pi x_3 - \pi) \text{ if } x_1 = 5 \end{cases}$$



Sandia National Laboratories

Test Functions 3

$$y = f_3(x) = \sum_{i=1}^{n} (x_i - 1)^4$$

- **Simple, analytic function**
- **We initially started with four variables:**
  - 2 continuous on [0,2]
  - 2 discrete with value [0,1,2]
- **Easy to scale up in terms of number of levels**
  - Scaled up the number of levels to five [-1,0,1,2,3].
- **Easy to scale up in terms of number of discrete variables**
  - Scaled up to five discrete variables, with three and five levels
- **Can also explore symmetry and function separability**

Polynomial Generator

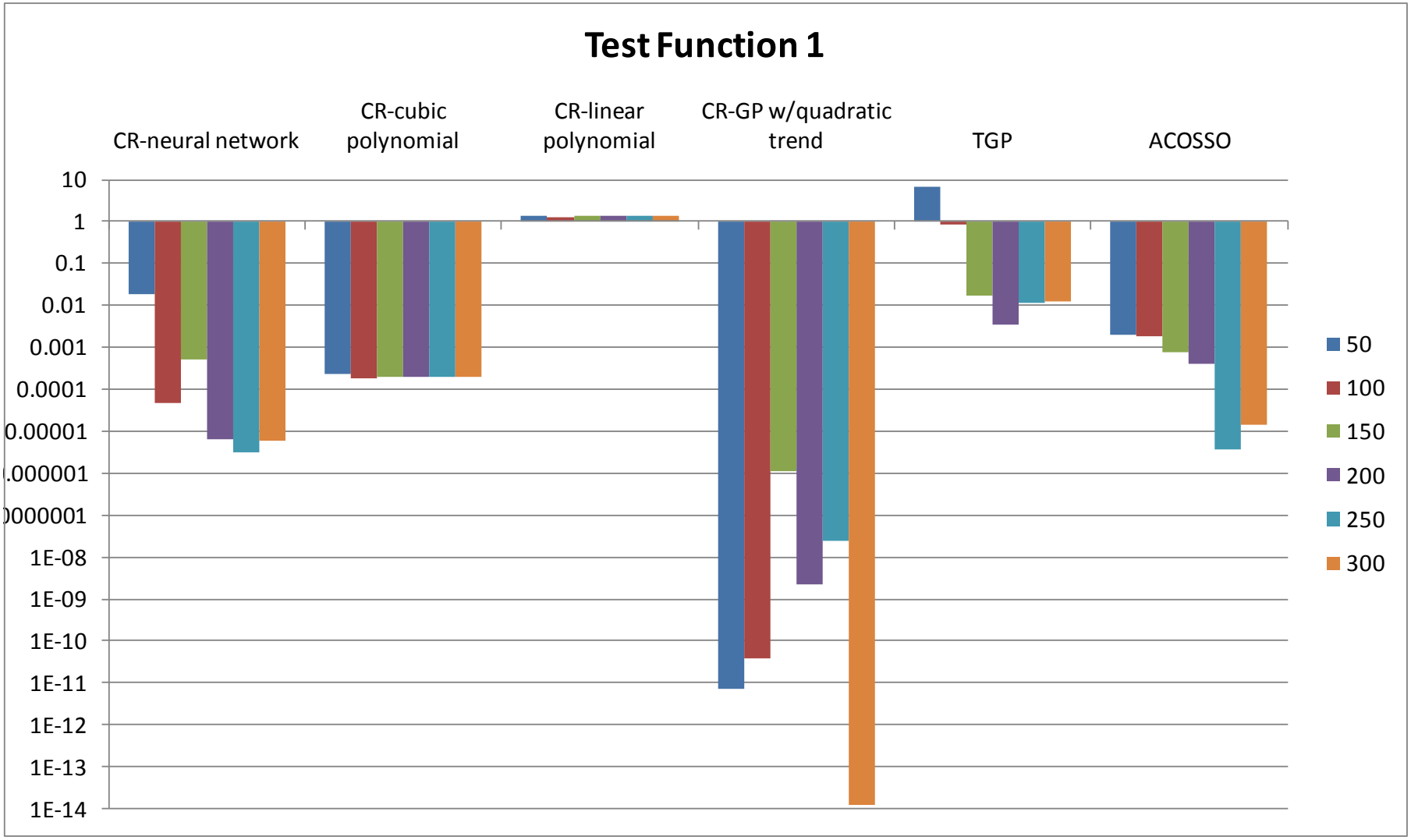
- **Generates a random polynomial with degree between two and six, and number of variables between one and fifteen**
- **Uses a polynomial generating algorithm which uses a system of linear equations to solve for the random coefficients, described in:**
  - McDaniel, W. R. and B. E. Ankenman, “A Response Surface Test Bed.” Qual. Reliab. Engng. Int. 2000; 16: 363–372
- **Can control the degree of nonlinearity, range of polynomial values, various features, etc.**

Evaluation Process/Scripts

- **For each surrogate type, looked at performance over a number of build points (LHS sample points).**
- **Used MSE (mean squared error) as a measure of goodness. Calculated MSE over a grid (where the grid was dimensioned based on the number of inputs).**
- **Categorical Regression run in DAKOTA**
  - Generate a separate continuous surrogate for each combination of discrete variable values/levels
- **TGP and ACOSSO run in R**

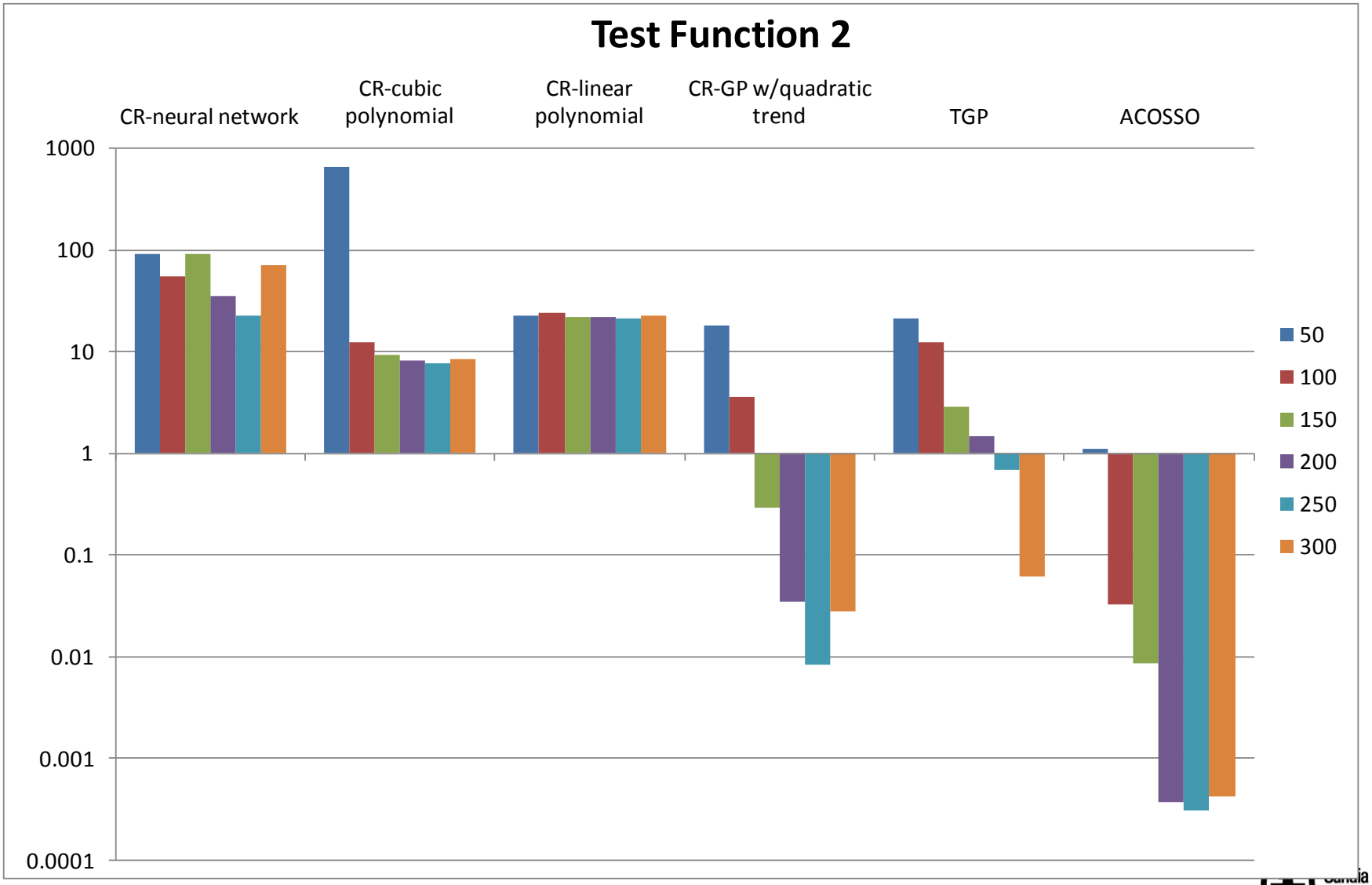
Results

Results: Test Function 1

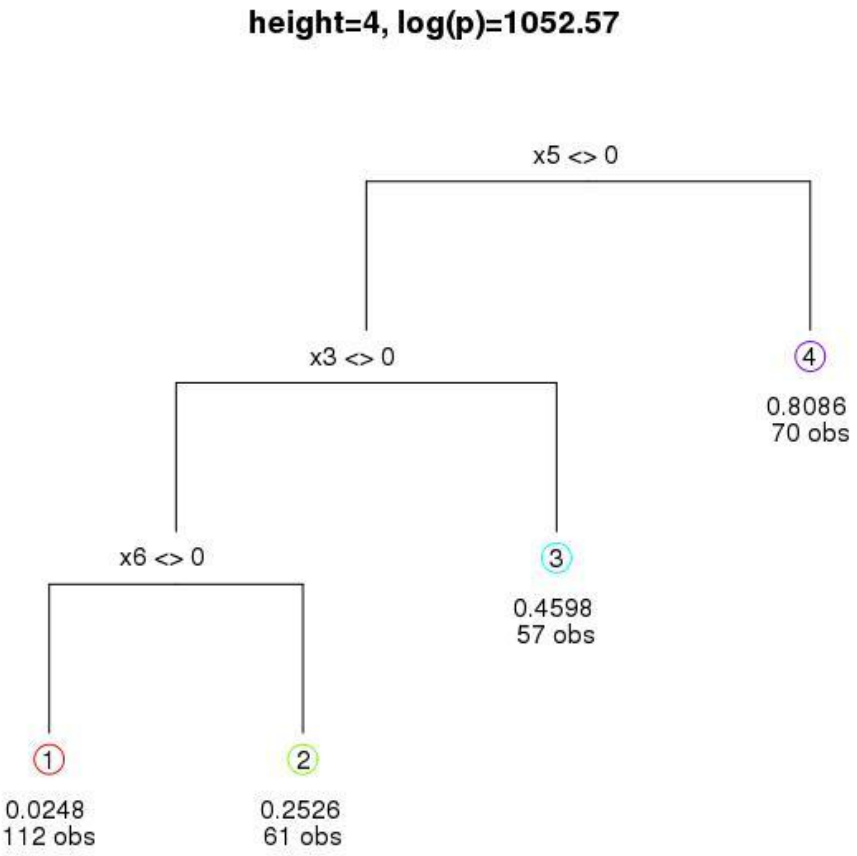


Sandia National Laboratories

Results: Test Function 2

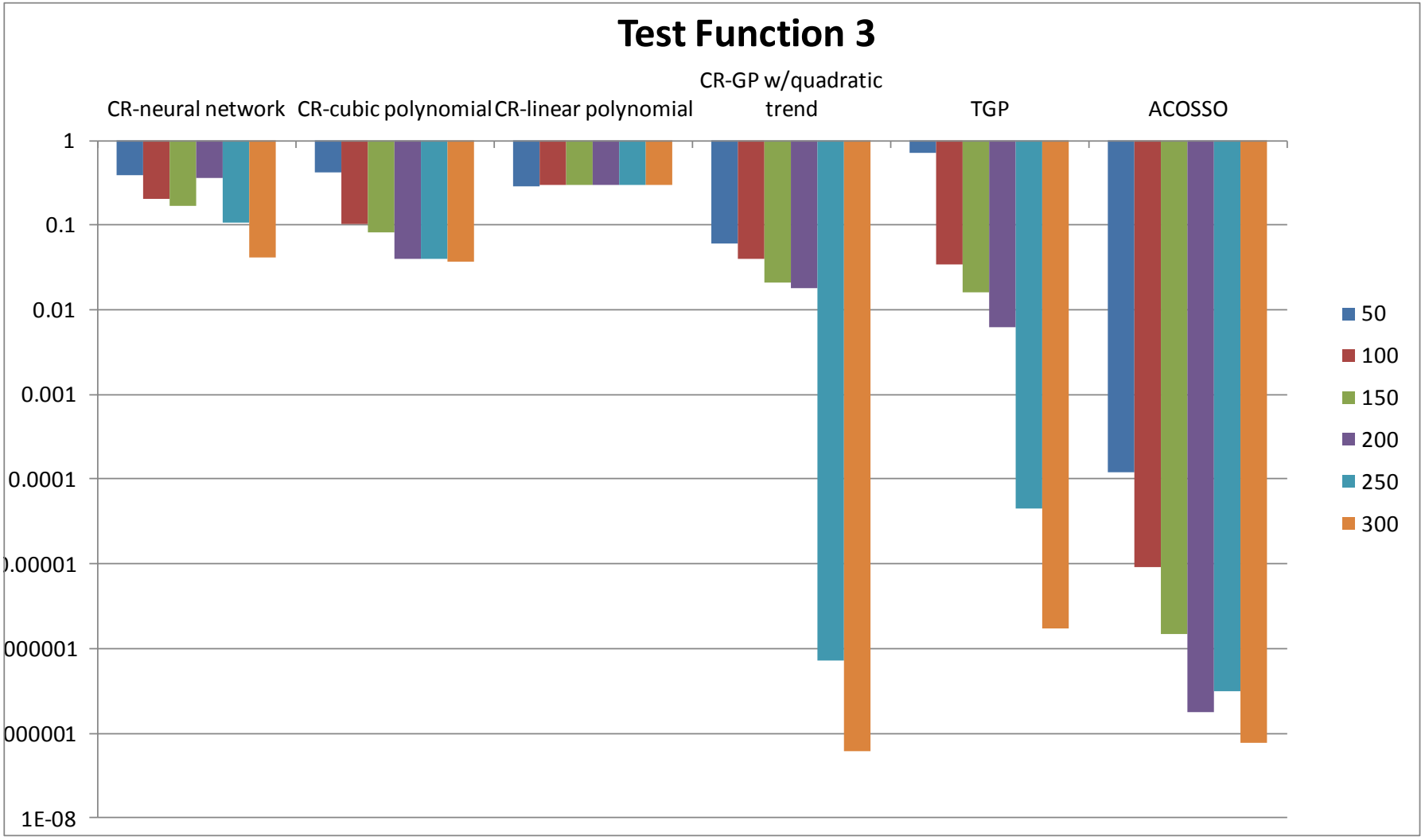


Results: Test Function 2



- The TGP does not fully partition over all of the discrete variables
- Our premise was that it would be sufficient to create surrogates over partitions which aggregrate the discrete variables (e.g. have one GP surrogate based on X1 at levels 1 and 2, and another with X1 at levels 3-5).
- It may be that this is too coarse, resulting in inaccurate surrogates.

Results: Test Function 3



Sandia National Laboratories

Results: Test Function 3, TGP

- SCALING UP DISCRETE LEVELS: FROM 3 to 5
- SCALING UP DISCRETE VARIABLES: FROM 2 to 5
- SCALING UP BOTH

Test Function 3	**SYMMETRIC - TGP**			
Discrete	2 [0-1-2]	2 [-1-0-1-2-3]	5 [0-1-2]	5 [-1-0-1-2-3]
Continuous	2 [0,2]	2 [0,2]	2 [0,2]	2 [0,2]
50	0.7217199	119.75	1.38	319.22
100	0.03391995	57.15	0.79	300.08
150	0.01617074	25.94	0.87	272.76
200	0.00631333	25.26	0.72	265.96
300	4.45E-05	17.91	0.52	231.41
500	1.74E-06	1.27	0.32	223.68

- MSE decreases more quickly for more discrete variables vs. an increased number of discrete level per variable

Results: Test Function 3, ACOSSO

- SCALING UP DISCRETE LEVELS: FROM 3 to 5
- SCALING UP DISCRETE VARIABLES: FROM 2 to 5
- SCALING UP BOTH

Test Function 3	**SYMMETRIC - ACOSSO**			
Discrete	2 [0-1-2]	2 [-1-0-1-2-3]	5 [0-1-2]	5 [-1-0-1-2-3]
Continuous	2 [0,2]	2 [0,2]	2 [0,2]	2 [0,2]
50	1.20E-04	8.15E-04	2.24E-04	2.56E-01
100	9.31E-06	1.55E-03	6.27E-06	4.06E-06
150	1.50E-06	1.34E-03	2.01E-06	2.56E-04
200	1.75E-07	3.20E-06	6.97E-07	1.99E-03
300	3.17E-07	4.68E-05	1.31E-07	5.24E-05
500	7.69E-08	3.08E-04	8.56E-08	2.14E-05

- MSE decreases more quickly for more discrete variables vs. an increased number of discrete level per variable

Results: Test Function 3, asymmetric case

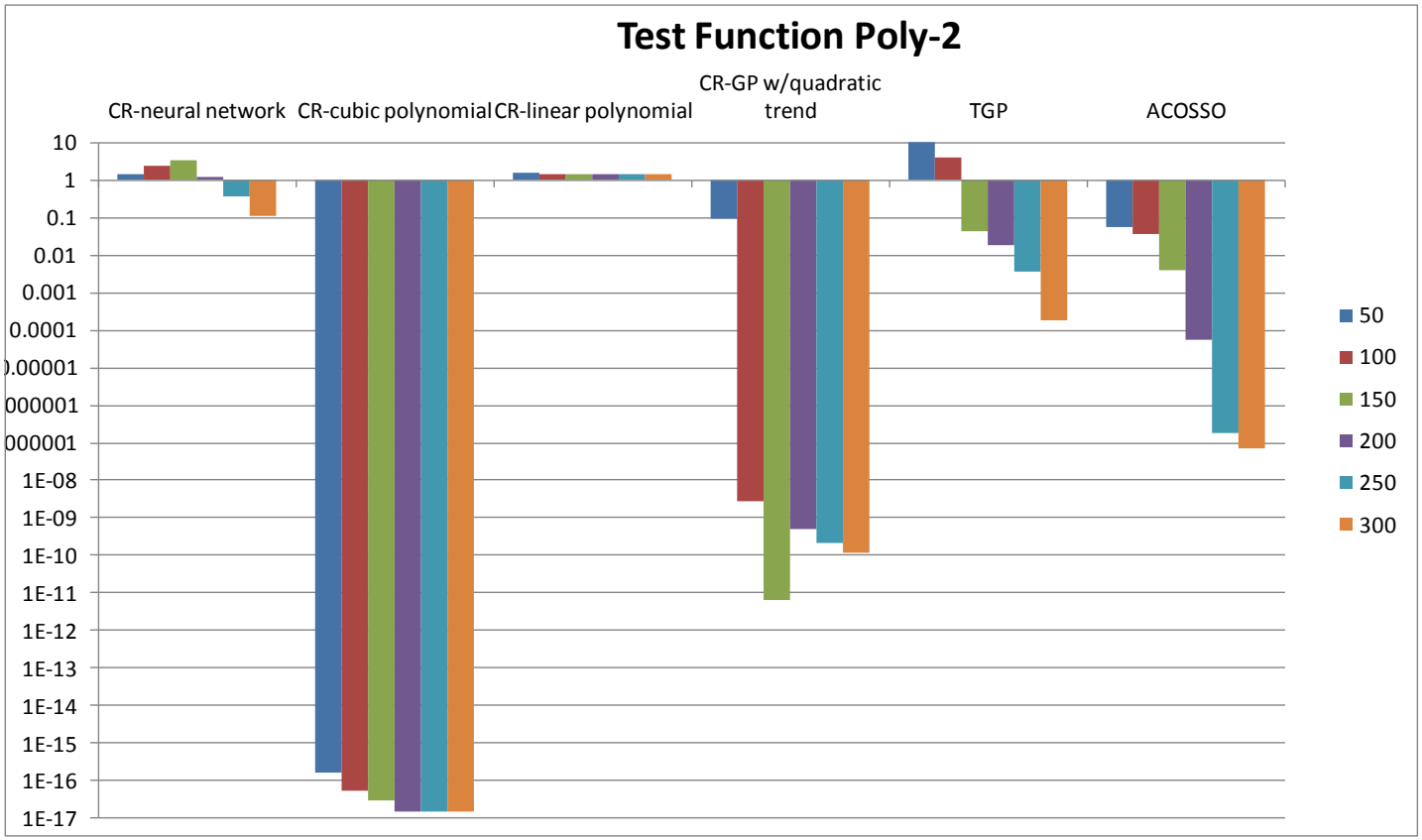
- SCALING UP DISCRETE LEVELS: FROM 3 to 5
- SCALING UP DISCRETE VARIABLES: FROM 2 to 5
- SCALING UP BOTH, but function is now ASYMMETRIC

Test Function 3	TGP		ACOSSO	
Discrete	2 [1-2-3]	2 [1-2-3-4-5]	2 [1-2-3]	2 [1-2-3-4-5]
Continuous	2 [0,2]	2 [0,2]	2 [0,2]	2 [0,2]
50	53.53	11843.35	3.67E-03	0.34
100	0.62	2444.52	1.96E-03	0.18
150	0.26	3402.73	3.27E-04	0.12
200	0.15	4494.97	9.94E-04	0.07
300	0.02	2382.42	3.55E-06	0.03
500	0.01	0.39	5.78E-04	0.06

- Asymmetry greatly increases the difficulty of the emulation for TGP, since symmetry helps by allowing different discrete levels to be binned into the same bin

Results: Test Function Polynomial

- 2nd order polynomial with four variables (2 discrete and 2 continuous)
- 2 discrete variables at levels [20,50,80]
- Continuous variables between 0 and 100



Sandia National Laboratories

Results: Test Function Polynomial

Problem Complexity: going from a 2nd order polynomial with 14 terms to a 3rd order polynomial with 24 terms to a 4th order polynomial with 19 terms

Finer discretization of discrete variables: 10 levels instead of 3

	Test Function Poly 2		Test Function Poly 3		Test Function Poly 4	
	TGP	**ACOSSO**	**TGP**	**ACOSSO**	**TGP**	**ACOSSO**
Discrete	2 [ten levels]	2 [ten levels]	2 [ten levels]	2 [ten levels]	2 [ten levels]	2 [ten levels]
Continuous	2[0,100]	2[0,100]	2[0,100]	2[0,100]	2[0,100]	2[0,100]
50	28.88	12.24	25.16	9.12	10.07	29.98
100	28.58	0.46	25.00	4.92	10.25	5.14
150	27.83	0.15	21.94	2.31	13.16	6.03
200	21.80	0.05	16.31	1.91	9.99	5.03
250	24.92	0.06	11.58	2.37	10.40	5.16
300	22.78	0.03	9.49	1.77	8.76	5.20

Observations

- **Categorical Regression performed very well on problems with small numbers of discrete variables/levels**
- **ACOSSO performs very well.**
- **TGP performance is mixed**
  - Functions 1, 2: performs well when it seems to get enough function evaluations (few hundred)
    - Ability to identify the splits
    - Not sufficient to aggregate across discrete levels
  - Functions 3, poly: performs poorly (i.e, high MSE)
    - There are adaptive methods, such as adaptively adding points to the GP based on expected improvement.

Observations

- **Scalability**
  - ACOSSO seems the most scalable, TGP suffers from too large an aggregation across discrete levels, and categorical regression is not scalable
  - Is there a difference between scalability across discrete variables vs. number of levels? Test function three suggests there might be
- **Further work:**
  - Amount of interactions between variables
  - Range/nonlinearity of function (polynomial testbed will allow)
  - Improving efficiency of TGP and ACOSSO implementations

References

1. R.B. Gramacy and H. K. H. Lee. "Bayesian treed Gaussian process models with an application to computer modeling." *Journal of the American Statistical Association*, 103:1119-1130, 2008.
2. R. B. Gramacy and H. K. H. Lee. "Gaussian processes and limiting linear models." *Computational Statistics and Data Analysis,* 53:123-136, 2008.
3. R. B. Gramacy and M. Taddy. "Categorical inputs, sensitivity analysis, optimization and importance tempering with tgp version 2, an R package for treed Gaussian process models." R manual available at http://cran.r-project.org/, 2009.
4. McDaniel, W. R. and B. E. Ankenman, "A Response Surface Test Bed." *Qual. Reliab. Engng. Int.* 2000; 16: 363–372
5. P. Qian, H. Wu, and C.F.J. Wu. "Gaussian process models for computer experiments with qualitative and quantitative factors. " *Technometrics,* 50(3):383–396, 2008.
6. B. J. Reich, C. B. Storlie, and H.D. Bondell. "Variable selection in Bayesian smoothing spline ANOVA models: Application to deterministic computer codes." Technometrics, 51, 110-120, 2009.
7. C.B. Storlie and J.C. Helton. "Multiple predictor smoothing methods for sensitivity analysis: Description of techniques." *Reliability Engineering and System Safety,* 93(1):28–54, 2008.
8. C.B. Storlie, L.P. Swiler, J.C. Helton, and C.J. Sallaberry. "Implementation and evaluation of nonparametric regression procedures for sensitivity analysis of computationally demanding models." *Reliability Engineering and System Safety,* 94 (2009) 1735–1763
9. C.B. Storlie, J.C. Helton,, B. J. Reich, and L.P. Swiler. "Analysis of Computationally Demanding Models with Qualitative and Quantitative Inputs." Draft manuscript.

Backup Slides

Why do we need surrogates?

- **Optimization and UQ methods (both aleatory and epistemic) are computationally expensive**
- **People want to perform a limited number of "true" simulations, construct surrogate, and perform analysis on the surrogate**
  - Engineering analysis: discrete choices for design options, physics "knobs" in codes
  - Logistics, SoS applications
- **There are a number of surrogates for continuous variables**
  - Parametric regression (e.g. linear, quadratic, rank)
  - Nonparametric regression (e.g. local regression, ridge regression, etc).
  - Splines
  - Neural Networks
  - Radial Basis Functions
  - Gaussian Processes
  - Stochastic expansion methods (e.g. polynomial chaos, stochastic collocation)

Why are discrete variables hard to model with surrogates?

- **Most surrogate models assume continuous inputs, and rely on some assumptions about how the output varies as the input varies**
  - In linear regression, the output is a linear function of the inputs
  - In Gaussian processes, the assumption is that outputs of input points "close together" will also be close together (governed by a correlation structure)
  - With discrete variables, this no longer holds (especially if the variables are categorical vs. ordinal)
    - E.g., changing from 3 depots to 4 depots may result in fundamentally different behavior of a logistics system
    - Model A vs. Model B vs. Model C may result in very different behaviors

Possible Options for Modeling Discrete Variables (2)

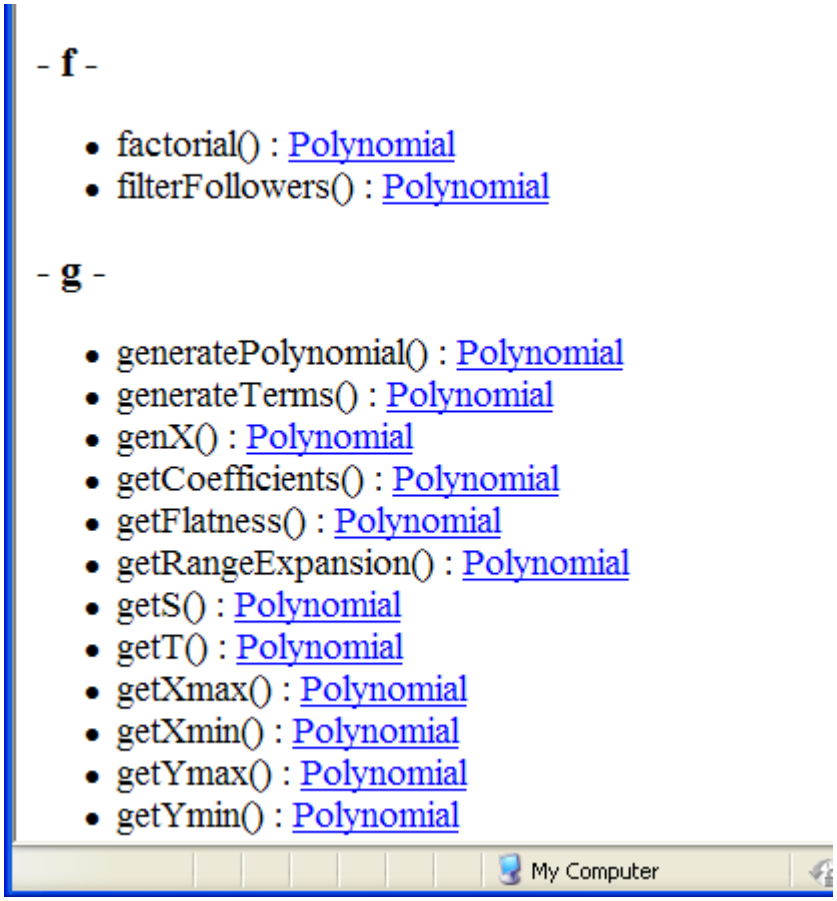
- **Gaussian Process Models**
  - Responses at points close together in input space are highly correlated, responses at points far away are not
  - Requires significant computational effort in determining parameters governing the covariance function
  - Can provide an estimate of the prediction uncertainty which can be used in optimization
- **Explicit Representation of Categorical Variables in GP**
  - Qian and Wu, 2007
  - Recommend isotropic correlation structure:

$$C(\boldsymbol{x}, \boldsymbol{x}') = v_o \exp\{-\sum_{u=1}^{d} \rho_u^2 (\boldsymbol{x}_u - \boldsymbol{x}'_u)^2 - \sum_{j=1}^{J} \theta_j I[z_{j1} \neq z_{j2}]\}$$

- **Treed Gaussian Process**
  - TGP R papers, Gramacy 2009, Gramacy and Taddy, 2010
  - Allow partitions on categorical variables
  - Surrogate model made at "leaf" nodes is formed only on continuous variables
  - Options: Bayesian GP with Linear Limiting Model

Testbed

- We developed a simple, extensible testbed, written in C++
- It currently has four test functions plus a random polynomial generator
- Fast running, uses SmartArrays
- Class documentation provided via html pages:



Testbed

- **Execution is simple:**

  tf –i input_file –o output_file –f fn#

- **Example**

  tf –i input_values –o output_values –f 1

input-values:

Function 1 input

X1=1

X2 = 0.42

output_values

2.50737536

```
Numterms: 14
..... Pure terms .....
73.9862
-0.279989 {1,1}
0.000350121 {1,2}
-0.136502 {2,1}
2.51957e-05 {2,2}
-0.242437 {3,1}
-0.0236206 {4,1}
-0.00130055 {4,2}
..... Mixed second order terms ....
0.000311869 {1,1} {2,1}
0.00217202 {1,1} {3,1}
-8.07668e-05 {1,1} {4,1}
0.00212612 {2,1} {3,1}
0.00171962 {2,1} {4,1}
0.00018751 {3,1} {4,1}
```

Testbed

- **Need a testbed for evaluation of these various methods**
- **Desired characteristics of a testbed:**
  - Fast running evaluations
  - Easy to compile, cross-platform compability
  - Extendable
  - File input/output
  - Scalability of function in terms of number discrete variables and/or levels per variable
- **Existing optimization testbeds not exactly appropriate**
  - Many tests either all continuous (e.g. to test quadratic programming, linear programming algorithms) or all integer (to test integer programming)
  - BARON does have MINLP problems, but the interesting part of the problem is constraints: we only need objective function

Results: Test Function Polynomial

- Scalability: going from 3 discrete levels to 10 discrete levels

Test Function Poly 2	TGP	TGP	ACOSSO	ACOSSO
Discrete	2 [20,50,80]	2 [ten levels]	2 [20,50,80]	2 [ten levels]
Continuous	2[0,100]	2[0,100]	2[0,100]	2[0,100]
50	9.19	27.14	0.36	10.79
100	8.34	25.61	0.05	0.22
150	0.12	25.04	0.05	0.18
200	0.01	24.26	0.03	0.16
300	0.01	18.04	0.04	0.11
500	0.00	10.81	0.03	0.09

- Problem Complexity: going from a 2nd order polynomial with 14 terms to a 3rd order polynomial with 24 terms

Test Function Poly 3	TGP	TGP	ACOSSO	ACOSSO
Discrete	2 [20,50,80]	2 [ten levels]	2 [20,50,80]	2 [ten levels]
Continuous	2[0,100]	2[0,100]	2[0,100]	2[0,100]
50	11.30	24.83	2.52	7.95
100	4.30	24.68	0.67	5.13
150	2.46	23.21	0.38	2.70
200	0.98	20.79	0.42	57.71
250	0.89	9.68	0.37	1.25
300	1.21	9.49	0.45	1.33